Who do you think you are?

A privacy-sensitive method for targeting online audiences

Patrick McCarthy
patrick@dstillery.com
Dstillery
New York, New York

Melinda Han Williams
melinda@dstillery.com
Dstillery
New York, New York

Wickus Martin
wmartin@dstillery.com
Dstillery
New York, New York

Jason Kaufman
jkaufman@dstillery.com
Dstillery
New York, New York

ABSTRACT

It is increasingly common when working with Internet data that records and statistics are not available on a record-by-record basis, but rather as aggregated summary statistics for a group of records. Although reasons for this vary, including provision of individual privacy or the desire to protect proprietary secrets, the absence of individual-level labels prevents direct application of conventional machine learning. In this work, we describe a two-step procedure combining ordinal classification on aggregate labels with an active learning-based approach to learn an individual-level classifier from aggregate labels. We evaluate our approach on the publicly-available MNIST digits dataset, and demonstrate significant improvement over a baseline approach for the multi-class prediction problem. Although class proportions (and therefore initial aggregate labels) of each digit are approximately 10% we show that with sufficient active learning iterations we can train single class classifiers to a range of 80-90% accuracy, and also a multiclass classifier to an accuracy of more than 60%. We discuss the application of our method to our own business, learning demographic labels for targeted advertising, and more generally as a practical approach to the use of machine learning in an environment with differential privacy guarantees.

CCS CONCEPTS

• **Information systems** → **Online advertising**; • **Computing methodologies** → **Active learning settings**; **Modeling methodologies**.

KEYWORDS

classification, active learning, online advertising

1 INTRODUCTION

Since the early days of the public Internet advertisers have endeavored to refine their ability to reach the most receptive prospective consumers. One approach to this challenge is the practice of audience-based targeting. Companies such as Dstillery gather data on user interactions with websites, using that data to train machine learning models which are ultimately used to segment web browsing devices into highly-targeted "audiences", each with a high propensity toward a particular interest or demographic characteristic. A typical approach from a Dstillery perspective is to train a classifier upon deterministically-labeled data to predict an outcome, such as purchases of a particular product. This approach is quite effective, however the transmission and storage of such deterministic user-level data may have undesirable privacy implications. In the name of responsible data science, one may seek to build a classifier in such a way that probabilistic propensities toward a particular action or trait may be learned and predicted without any access to private individual-level ground truth label data. We envision several scenarios with these constraints. A company with access to a great deal of data about individual behavior may wish to limit liability and reassure clients by making sure no system, external service provider, or internal employee or system can look at single records. To this end, the company may insert an automatic aggregating layer between raw data and its consumers. Alternatively, a provider of a rich dataset or measurement service may be motivated to protect trade secrets and proprietary information, and so may only release an aggregate summary to their subscribers. A data owner may even aggregate event-level data such as website visit statistics in order to report important trends efficiently and succinctly.

Suggested responses to this dilemma have taken various forms. k-Anonymity [8] describes a framework in which any individual observation in a dataset is indistinguishable from at least k others. Another approach, ϵ-δ Differential Privacy [1], provides a series of rules which prevent the identification of an individual from aggregated data when other observations in the dataset are already known. These approaches and others may be implemented in such a way that complete data is replaced by an aggregate representation.

Although doing so presents an obvious benefit both to an individual (who does not want to reveal her data) and the owner of a dataset (who may wish to protect trade secrets or proprietary information), the machine learning practitioner faces a challenge. How can we use aggregated data when the mass of modeling algorithms rely upon specific labels tied to specific features?

For the purposes of this work, we focus our attention on the following scenario. We wish to solve a multi-class classification problem in which each individual belongs to one of a set of mutually-exclusive classes. For whatever reason (*e.g.* privacy, secrecy) ground truth class labels are only available in aggregate, but at the same time aggregate labels may be requested multiple times and are available for any sufficiently large sample of individuals. In this scenario, an active learning[9] strategy allows us to request specific data which we know may bolster our understanding of the problem. Active learning is often understood to describe a situation in which

a query is answered with complete data, but it may also provide an obscured or summarized response. Such a system that makes such indeterminate proclamations, perhaps without revealing the nature of its sources or rules of response, is referred to as an "oracle".

To the extent that we have an oracle which can provide us with some insight, we would like to learn from this oracle so that we ourselves may learn a relationship between the Internet activity data and these real-world characteristics of interest. Given that we have a series of groups of observations, each one assigned a single aggregate label by the oracle, we could attempt to learn from the differences between the labels, grouped though they are.

It is difficult to learn a higher in-target proportion (that is, a higher value for a label) than can be observed in our data. Many textbook machine learning algorithms are unhelpful in this regard, particularly those which find the expectation or posterior mode of a label conditional on the data — even if perfectly accurate, no guarantees are made that the resultant predictions are stable beyond the domain of the training set, which is precisely our goal. We address this problem by resampling our data to increase variance in the labels available to us, and by training an ordinal classifier, a model which learns how to rank predictions into quantiles.

To find groups of observations which become successively more in-target, we create a process of "ratcheting" by applying an active learning approach: we select a number of groups from our dataset universe, submit them to the oracle which returns their aggregate labels, and leverage what we can learn from those labels to arrange observations into a new set of groups, which themselves are submitted to the oracle and so on. This empowers us to discover "only the good stuff" from each iteration, effectively sorting our observations and bringing the aggregate label successively higher. With incremental improvement between successive generations of segments, it becomes possible to start with an effectively random sample of observations which the oracle may say are 5-10% in-target for a desired characteristic, and develop groups for which the proportion of observations with that characteristic is much closer to 100%.

In this paper we present a means by which a predictive learning task may be performed on labels aggregated by an oracle. We use the oracle to obtain the aggregate characteristics of a series of random samples of our dataset, and by training a classifier to identify those samples which have the highest value for a desired label we can improve our chances of drawing even higher proportion samples in the future, thus bootstrapping our way to the selection of more homogenous groups. We then make use of these homogenized populations to train a conventional multiclass classifier for purposes of predicting the class memberships of individual records. We describe a reproducible simulation of our method applied to a common dataset, and describe our efforts to apply the method to targeted display advertising in practice.

Our contributions to the literature are as follows. 1) To the best of our knowledge our method is the first proposed which produces an accurate prediction of a minority class using only aggregate, proportional ground truth labels. 2) We demonstrate a novel two-stage method where each active learning iteration is not used to train a final model, but to gather increasingly refined training data, all of which labeled by an authoritative source. 3) We demonstrate our approach reproducibly through a demonstration implementation upon public data, released on GitHub. 4) Finally, this work contributes to a developing understanding in the industry of how to work within an environment defined by intentionally obscured ground truth data. We find it important to note that our method is not intended to reverse-engineer oracles *per se* but to work alongside them — we respect the desire of data owners to manage what and how they share, and wish to demonstrate effective data science practice within these confines.

2 PROBLEM DEFINITION

Consider a dataset X composed of a series of n records, each composed of a feature vector $x^{(i)}$ of length p. Each record describes a single individual in a population although the data set may, via duplicate records, describe fewer than n individuals without loss of generality. Although we are in possession of X in its entirety, there also exists a label vector y which describes some property of the individuals which is totally unknown to us. Each value $y^{(i)}i$ in y takes one of a set of c discrete labels $\{v_1, v_2, \ldots, v_c\}$ and so assigns every observation $x^{(i)}$ to exactly one class. Our objective is to train a classifier such that, for a new individual record $x^{(n+1)}$ we can correctly predict the value of $y^{(n+1)}$.

In our scenario we presuppose the presence of an oracle like that described by Valiant [10]. The oracle knows the true values of the labels y, and when queried about a subset of records $q \in X$ it responds with corresponding label information y_q. Unlike Valiant's oracle, we also suppose that ours is both unwilling to disclose individual results (*i.e.* that it will only respond to queries of some cardinality $>>$ 1) and that what responses it does provide are aggregated. For each query, an aggregated response consists of c values, describing the proportion of the observations in q belonging to class v_1, the proportion belonging to v_2, *etc.*

We begin from an initial state in which we have no prior information about y, and so any subset of $q \subset X$ is effectively a random draw from X.

3 RELATED WORK

The problem of learning without clear labels is not a new one, with much work done in the online marketing space. Often, these studies seek to make use of a third party demographic scoring service in concert with their own data to predict demographic characteristics. Williams *et al.*[11] address the question of casting aggregate labels into a binary classification problem to predict demographic characteristics for online advertising. In that work, devices from different segments were submitted to an oracle anonymously and at random which motivated two separate labeling schemes. As probabilistic labels, the aggregate label of the segment was used as the parameter of a Bernoulli random draw, assigning either a 1 or 0 to each observation. By contrast, maximum a posteriori (MAP) labeling considered the oracle's assigned aggregate label to be a reflection of the probability of membership in a particular class to the exclusion of others, and so assigned a 1 to the class with largest probability and 0 otherwise. As in the introduction, we found these methods unsuitable for a multiclass case where there is no clear "majority" class, such as in the case of an evenly-distributed ten-class problem.

In a work with similar ambitions, Renehan[7] approaches the problem of predicting demographic labels not as a consumer of Internet browsing history, but by looking at the apps installed on

mobile devices. This study began with a number of "seeds", weakly accurate predictors of demographic data. Devices appearing in the same demographic bucket across multiple seeds are considered promising prospects, and this pool is then winnowed down by training a model on these prospects and then predicting the class membership of same. Observations unable to be correctly classified by the model were removed, "purifying" the data. A second model is trained upon this subset of devices and evaluated on their whole universe of devices, yielding predictions to inform their segments. Although his method shares our practice of scoring a whole dataset universe with a trained model to find new in-target records, it does not progressively sample the oracle to validate its predictions nor does it combine these predictions to train a multi-class classifier.

Kar *et al.*[3] take a slightly different approach, using demographic mix data reported on the viewership of a television show. Using a Bayesian framework, they develop a two step classifier in which first prior probabilities for membership in a demographic class are computed on a vector composed of television shows watched. These priors are then combined with features describing online activity, and used in a logistic model to predict class membership. They too, like Renehan, define a method without active learning as they have aggregate labels but not the ability to request further information on subsets of their data.

4 METHODS

4.1 Concepts and Terms

In our domain of online advertising, our objective is to attempt to predict behavior and characteristics so as to best match a future advertising opportunity or "offer" with the most receptive possible individual. This activity is typically captured at the level of a web browser or mobile device, and reflects our awareness of the web pages visited by the device, perhaps aspects of particular web pages (*e.g.* if it is a page showing a product or a page congratulating an individual on a purchase) and associated metadata, including the time at which these visits occurred and technical information describing the device. Together, we consider the device's data to be an "observation" i, and the combination of its visit activity and metadata to be its "features", the vector $x^{(i)}$. These data are often used in machine learning models to best predict future actions to be taken by the individual as well as topics which may be of interest to her.

In order to make use of as many advertising opportunities as possible, advertisers must be ready to show an ad to a receptive individual whenever one appears and yet at the same time not overwhelm her by continually showing her the same ad. Additionally, we cannot predict when a particular individual may visit a site and make herself available to be shown an ad and so the most must be made of what opportunities present themselves. To this end, marketers will purchase access to groups of devices which have been determined to share interests or histories which are called "population segments" or simply "segments". It is by grouping observed devices into segments and then targeting different segment combinations that advertising campaigns may define their audiences.

In order for these segments to be useful, it is assumed that a significant fraction of these segmented devices will be observed again at a later date. This allows us to study sequential behavior (*e.g.* "many people will visit a reviews website immediately before visiting a restaurant website") as well as to configure our system to take an action when they appear, such as forwarding newly learned information to a partner firm. An oracle may be the recipient of such a forwarding action, which allows us the great advantage of selecting which devices we would like our oracle to measure. This technique, in which we can query an authority for additional information about some data is called Active Learning. We do not require our oracle to issue its response based upon any data in our possession, or even an observation ID known to us, but rather we can only forward a returning device to the oracle which will act on its own private information. As such, the ability to pick and choose which different elements of our dataset universe will be sent into the oracle is central to our method.

For organizational purposes, we store our data in two different places. We consider our "dataset universe" (or simply "universe") to consist of all distinct observations and features known to us at a given time. Though the number of observations in the universe may increase or decrease, we assume no agency over the members of the universe and so consider it fixed. By contrast, we define the "data store" to be a database consisting of those segments which have been sent to the oracle, the observations they contain, and the aggregate labels received. We manage the contents of the data store in that we select the segments submitted to the oracle, and as it is our sole repository for the labels we received we use the data store as the source of all model training data.

4.2 Techniques

Following the example of Williams *et al.* [11], we wish to find a practical means of connecting the aggregate label assigned by an oracle to each of the observations in the group from which the label was aggregated. Unlike that work, however, we don't share the requirement of producing a method compatible with logistic regression. The approach that follows will take the following form: first, we develop a single-class classifier which predicts a label for class v_i. We then apply this classifier in an active learning context, refining our ability to make accurate predictions and developing the ability to define segments in which predicted labels $\hat{y}$ approach 1.0. Finally, with developed classifiers for classes $\{v_1, v_2, \ldots, v_c\}$ we can leverage new, homogenized segments to train a single multiclass classifier which may make predictions for individual observations.

4.2.1 Single-class Classification. Consider an OLS regression problem in which a vector of coefficients $\hat{\beta}$ is learned by finding $argmin \sum_{i=1}^{n} (x^{(i)}\beta - y^{(i)})^2$. (Going forward we assume all parameters will be found via numerical methods and so sidestep potential issues of non-invertibility, *etc.*) We can slightly re-express the objective function to be $(x^{(i)}\beta - \mathbb{E}(y^{(i)}))^2$, where $\mathbb{E}(y^{(i)})$ is the average over a single observation. We may take the expectation over subsets of y larger than one — this diminishes the ability of our regression to learn optimal parameters but does not eliminate it entirely as its tantamount to injecting an error term into our predictions: in a subset of size 2 we now must find $\hat{y} = \mathbb{E}(\mu_y|X)$ which decomposes into $\hat{y} = \mathbb{E}(y^{(i)}|X) + \mathbb{E}(y^{(i+1)}|X)$. Although the additional variance introduced by the $y^{(i+1)}$ term reduces the information available to the algorithm, Williams *et al.* explored similar approaches with

some success. Additionally, learning regression coefficients from expectations was explored in depth by Quadrianto[6].

We refer to a series of r segments $s_1, s_2, \ldots, s_r$, where each segment s_i is composed of $n_{s_i} > 1$ observations. We concatenate these observations into a training set X, but because we rely on aggregate labels from the oracle the labels have cardinality $\leq r$. This is the generalization of the method just described, however instead of sharing the label over two observations we now do so by sharing each segment label y_s among all the observations $x \in_i$. We now have a dataset consisting of the label vector y of length $\sum_1^r n_{s_i}$ and cardinality r, and the feature matrix X with dimensions $(\sum_1^r n_s \times p)$ where p is the length of $x^{(i)}$. These data are now of the appropriate shape to accommodate OLS regression. Because of the straightforwardness of this approach, we will refer to this as "Naïve Regression".

Although naïve regression is useful, its reliance on expectations doesn't allow for the stable ratcheting effect we need to transform segments from a small proportion of our desired class to a larger one. We must distinguish here between two perspectives on expectation: expectation of the label in the data is unavoidable and forms the basis of our scenario. Separately, linear regression produces the expectation $\hat{y}$ of the label on the data and so conditional on data attempts to find a mean among the labels from which it learns. In essence, our procedure requires us to predict well on data beyond the range of our training set, and OLS is not expected to perform well at this task. To this end we make use of an ordinal classifier as specified by Frank and Hall[2]. To be explicit, the purpose of this classifier in our method is to bin the aggregate label y into discrete quantiles and to classify observations into the correct bin, distinguishing it from the multiclass problem described below. Rather than estimating parameters for all training data and labels at once, this approach requires decomposing the model into a series of binary classification problems. First, the domain of labels y is divided equally into q quantiles, separated by a set of $q-1$ cutpoints $k_1, k_2, \ldots, k_{q-1}$. At a cutpoint k_i, a new label y'_{k_i} is defined to be 0 where $y < k_i$ and 1 otherwise. For each such cutpoint, a binary classifier is fit and class probabilities are predicted for all members of the training set. Combining the probabilities of each model and subtracting between them, for each feature vector $x^{(i)}$ we compute a class probability vector $w_{x^{(i)}}$ of length q for which $\sum w_{x^{(i)}} = 1$. To borrow Frank and Hall's example, imagine one wishes to apply this method to outdoor temperatures in $\{Cold(< 5C), Cool(5 \leq x < 10C), Mild(10C \leq x < 20C), Hot(> 20C)\}$ we would first classify "cold" OR "cool", and then "cool" or "mild". As ("cold" or "cool") is the compliment of "mild", we know that "cold"'s contribution must be subtracted from ("cold" or "cool") as in Figure 1. Details may be found in the appendix as Algorithm 1. This algorithm has two distinct advantages — the individual classifications can be performed by any binary classifier capable of producing predicted class probabilities, allowing for flexibility of implementation. More importantly for our purposes, the binary classifiers are valid for any inputs above or below the cutpoint of interest. This allows us to reason about proportions (labels) higher than those that exist in our training set, enabling us to recognize and react upon segments with progressively higher in-target percentages than would the OLS-based naïve case.

Given classes *Cold, Cool, Mild, Hot* :

P_{cold} = 1 - *Pr(temp > Cold)*
P_{cool} = *Pr(temp > Cold) - Pr(temp > Cool)*
P_{mild} = *Pr(temp > Cool) - Pr(temp = Hot)*
P_{hot} = *Pr(temp > Mild)*

Cold	Cool	Mild	Hot
0.6	0.4		
0.85		0.15	
0.9			0.1

	Cold	Cool	Mild	Hot
	0.6			
	0.6	0.4 - 0.15		
	0.6	0.25	0.05 - 0.1	
class = argmax(	0.6	0.25	0.05	0.1)

Figure 1: The Ordinal Classification of Frank and Hall

4.2.2 Active Learning. Relying upon the ordinal classifier to predict the quantile to which each observation belongs, we may now make use of this method to produce more homogenized segments via active learning. By gathering data from our data store and sorting according to known aggregate labels, we may select a training set reflecting the highest and lowest labels for a class v_i. (We may additionally choose to add other observations from the center of the distribution to avoid a model overfit on extreme observations.) An ordinal classifier is fit and then used to score the whole of the dataset universe, and a selection of devices from the highest quantile are grouped into a new segment and submitted to the oracle for labeling. We may wish to submit several such segments for volume, the members of which are each randomly sampled from the prediction of the highest quantiles. As we scored our whole universe, it becomes possible to discover observations with a higher probability of being in the top score quantile than any in our training set. The oracle will return aggregate labels for all submitted segments, and for every segment the segment label is applied to each of its member observations, which is then re-inserted into the data store as a new entry. As the classifier develops, its ability to select desirable observations increases, causing the homogeneity of new segments to increase as well. This operation is summarized in Figure 2 and as Algorithm 2.

4.2.3 Multiclass Classification. After several iterations of the active learning operation, the aggregate labels of the oracle will show segment membership growing increasingly homogenous as the classifiers generating segments improve in discerning the characteristics in their intended class. After producing sufficiently uniform segments for each class $v_1, v_2, \ldots, v_c$ we may combine them to generate a multiclass classifier. We again draw observations from our data store, now selecting only observations for which the label is above a certain threshold t. Recall that this reflects the oracle's indication that the observation belonged to a segment which was $t\%$ in-target for our label of interest and so our observation with label t has a $t\%$ chance of being in-target itself. Converting these labels to crisp categorical or integer MAP labels as the multiclass algorithm

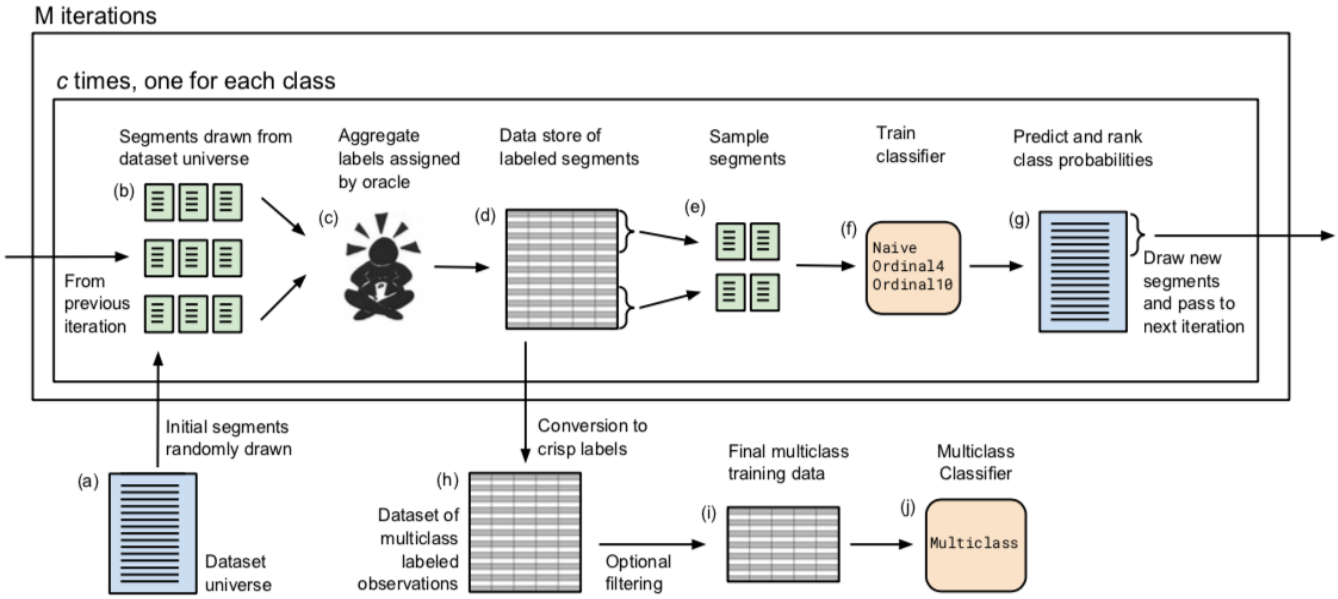

Figure 2: Active Learning and Classification. The full dataset universe (a) contains feature vectors for all available observations. The method is initialized by drawing random segments (b) of observations from the dataset universe. Segments are passed to the oracle (c) which assigns each segment an aggregate label, specifically the class proportion of the segment for the class in question. The segments, their corresponding individual observation feature vectors, and their aggregate labels are added to the data store (d). Segments (e) are then selected from the data store, according to sampling logic described in the text. Individual observations are assigned the class proportion labels of their segments, and this dataset is used to train the classifier (f), following one of three methodologies described in the text. The classifier is used to score all observations in the dataset universe (g), which is then ranked according to the predicted class probabilities. New segments are drawn from the ranked dataset universe, as described in the text, and used in the next iteration of active learning. After all iterations are complete, the data store (h) of labeled segments is converted to a data store of crisp, multiclass labeled individual observations, as described in the text. After an optional filtering step, the final training data (i) is used to train the mutliclass classifier (j).

requires, we may now train the multiclass classifier, concluding our method. The computation on refined segments producing a multiclass classifier is summarized in the appendix as Algorithm 3.

5 SIMULATION STUDY

To demonstrate the method we perform a multi-class classification on the famous MNIST Dataset [4]; the code of the demonstration may be found at https://github.com/asteriske/privacy_sensitive_active_learning. These data represent images of handwritten digits 0-9, and are composed of a label vector y for which each element $y^{(i)}$ is the corresponding integer in $\{0, 9\}$. The features X are composed of a 28x28 pixel map resized into a vector of 784 values, each of which valued between 0 and 255 to express how strongly the pixel is shaded. This dataset provides us with several advantages: it is commonly used and understood in the machine learning community, provides distinct mutually-exclusive multiclass labels, and is of a size such that samples may be repeatedly drawn without significant risk of inadvertent differential attack.

We retrieve the dataset using the `fetch_openml` function in Scikit-Learn [5], after which we divide it into a training set of 60,000 rows and a test set of 10,000 rows. We initialize our data store of training data by repeatedly drawing random samples with replacement from the training set, retrieving an aggregate label for each from the "oracle", and then storing the aggregate labels along with the observation IDs in a database.

To train a model for a particular label, the database is sorted according to the aggregate label of interest, for instance the proportion of sample observations labeled as "3". 100 samples are drawn with the highest in-target percentages, 100 more with the lowest in-target percentages, and for stability a third 100 selected from the database at random.

Recall that each sample is represented by a collection of observation IDs. To reconstitute a training set for the ordinal classification problem, the rows of data (that is, the 784-dimensional vector of pixel values for each image of a handwritten digit) for each observation ID are retrieved, and the observation label is assigned as the the sample's in-target percentage. Concretely, if a sample has four observations with true labels $\{1, 1, 1, 3\}$ and we wish to learn to recognize 3, then all four observations will be given the aggregate accuracy label 0.25.

The segment selection task is then performed, which in our scenario consists of three cases: the ordinal classification of [2] based on four quantiles, the same on ten quantiles, and naïve regression. The four-quantile ordinal classification procedure is as

follows. First a logistic regression is trained using Scikit-Learn's `SGDClassifier` to learn whether a label is below or above the 25th percentile of all values in the sample, then another to learn whether a sample is above or below the 50th percentile, and then finally one to learn above or below the 75th percentile. The class probabilities for each classifier are found for the whole universe (that is, the complete 60,000 item dataset), and after the necessary arithmetic the argmax of these probabilities is taken to assign each observation a predicted label indicating whether its probability of having a desired label (a "3") is in the first, second, third or fourth quartile. Among those found to be in the top 25%, several samples of observations are collected (again with replacement) and submitted to the oracle to provide a possible basis for a further iteration. The ten quantile approach is run in similar fashion, selecting new segments from the top decile. The naïve case experiment is also fit similarly, using Scikit-Learn's `SGDRegressor` and drawing samples for resubmission from above some configured score threshold. Both the classifiers and the regressor are trained for optimal penalization terms using three-fold cross-validation. All simulations are run for 50 generations on each class; each generation of a class re-inserts 10 segments of promising observations into the data store, ultimately adding 500 segments per class.

Once the iteration stage is done, data is drawn from the data store to fit a ten-class problem using `SGDClassifier` with hinge loss and elastic net penalty. The classifier is fed crisp categorical labels by, for each observation ID, selecting the highest aggregate label it received for each class and then of those the class with the maximum label is assigned as the observation's crisp label for this problem. If an observation ID in the data store appears in three segments which have labels $\{v_3 : .9, v_3 : .8, v_8 : .2\}$ then as 0.9 is the maximum value it would be assigned '3' as its label. Each observation ID therefore only appears in the multiclass training set once. The multiclass experiment is performed twice, once using every observation's maximum aggregate label, and again as a high confidence version in which observations with maximum aggregate label < 0.8 are removed. The multiclass problem is finally evaluated on the held out test set of MNIST digits, totally unused by any part of the process.

6 SUMMARY OF FINDINGS AND DISCUSSION

We examine the performance of the ordinal classification method against naïve regression from several perspectives, including the accuracy of newly-derived segments, rate of discovery of correct devices, and the performance of the final classifier.

6.1 Accuracy of New Segments

Recall that our process first requires new segments to be generated and inserted into our data store before being used to train further classifiers. At present Dstillery uses a method similar to the naïve approach in production, and as a large part our business involves curating accurate segments, an improvement in this task alone is advantageous to us. We analyze the paths of each class's segment accuracy from initialization to completion, and summarize segment accuracy by class after iteration 25, when all have been "burned in". Looking both at the in-target accuracy plots (Table 1) and the convergence tracks throughout the lifetime of the modeling process

Table 1: Segment Accuracy by Model Type (mean±sd)

target	Naive	Ordinal4	Ordinal10
d0	0.57±0.33	0.75±0.07	0.96±0.03
d1	0.45±0.24	0.91±0.04	0.99±0.02
d2	0.49±0.31	0.73±0.07	0.93±0.03
d3	0.33±0.22	0.72±0.07	0.93±0.04
d4	0.31±0.2	0.72±0.1	0.95±0.04
d5	0.53±0.28	0.73±0.06	0.91±0.07
d6	0.64±0.33	0.71±0.09	0.95±0.04
d7	0.41±0.25	0.82±0.06	0.97±0.02
d8	0.24±0.15	0.48±0.12	0.69±0.15
d9	0.25±0.16	0.63±0.07	0.83±0.14

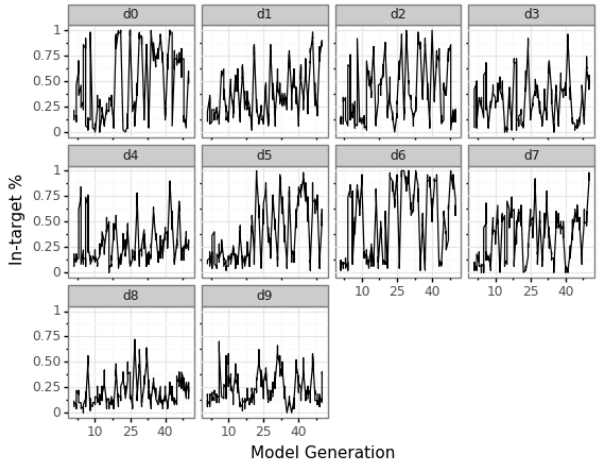

Figure 3: Naive: Segment Accuracy vs. Generation

(Figures 3, 4, 5), we see that the ordinal classifiers surpass the naïve classifier both in within-class accuracy and stability, characterized by the mean and standard deviation, respectively. The 10-class approach is particularly good, with some little additional stability.

The naïve approach is much less stable than either of the ordinal cases with all class generating segments which are close to or near zero accuracy throughout the entire process. We believe this to be a consequence of the limitations of OLS regression, particularly that the relationship being fit may not be linear for observations not in the feature set, or indeed linear at all. Of similar interest is the disparity in convergence plateaus between the 4- and 10-quantile ordinal classifiers. As many of the quartile classes peak near 75% while the decile classes can peak near 90%, there is an apparent and intuitive relationship between the size of the quantile buckets (and the variance of observations sorted into them) and the model's ability to create homogeneous segments.

Another useful diagnostic plot is that of the distribution of segment accuracy as it evolves over time (Figures 6, 7, 8). We see that our initial state is composed of random draws and so has a natural accuracy of 1/(# classes) or $\approx$ 10%. The ideal case sees this density move toward 100% as the procedure runs, and we see this borne out

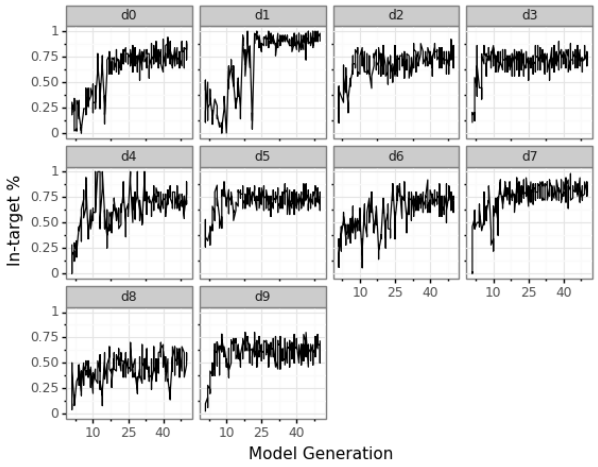

Figure 4: Ordinal4: Segment Accuracy vs. Generation

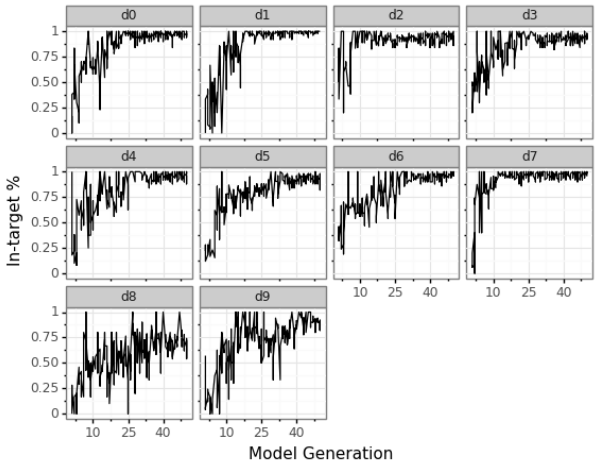

Figure 5: Ordinal10: Segment Accuracy vs. Generation

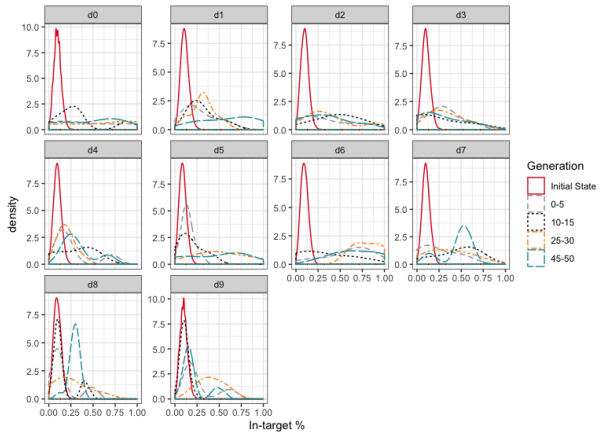

Figure 6: Naive: Segment Accuracy Across Training Steps

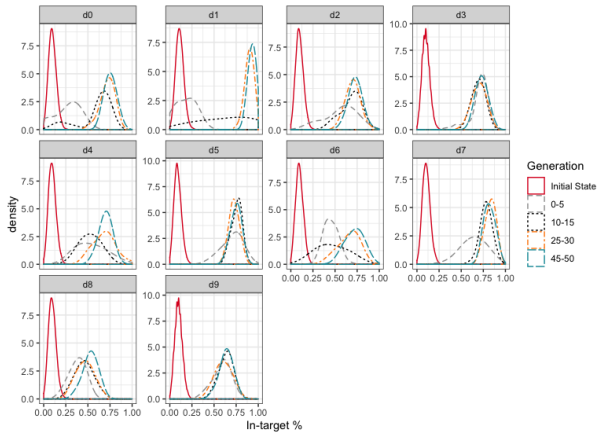

Figure 7: Ordinal4: Segment Accuracy Across Training Steps

in the four-class ordinal classifier and even more so in the ten-class case.

6.2 Observation Discovery

Although each method arrives at a degree of accuracy, it would be trivial to maintain some fixed level of accuracy over successive iterations by re-submitting the same observations to the oracle (and the dataset universe) again and again. Figure 9 illustrates the rate at which new observations are labeled by the oracle and added to our data store for the first time. To minimize distractions by random, poorly predicted observations we count only those new observations which appear in segments more than 50% accurate.

That said, it is also interesting to consider the part played by undesired observations in the segmenting process. Although it is tempting to focus on refining a somewhat-homogeneous subset of observations to be even moreso, the addition of out-of-target observations (those with low labels) and observations selected among the range at random allows not only for the classifiers at lower cutpoints to be more robust, but also to ensure we don't overfit on features endemic to our class of interest.

We wish to emphasize that although we are progressively optimizing classifiers, it is this discovery that is in fact the motivating application to our business. As we build new segments it is generally infeasible for an oracle to evaluate millions or tens of millions of observations, and its equally difficult to make use of new observations in our universe without designing a classifier for the purpose. In doing this, we achieve a substantial improvement to our business practices.

6.3 Multiclass Classification

Although falling far short of cutting edge MNIST digit performance (0.17% error rate at time of this writing[**?**]), all three classifiers performed remarkably for starting with no prior truth (Table 2). Breaking out performance into individual classes, there are some remarkable differences in strength between an "all observation" model and its high-confidence variant (Figure 10). Differences between models may owe to each experiment having a different train-test

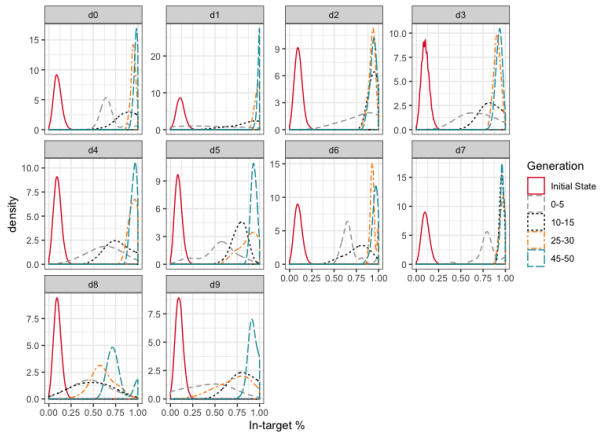

Figure 8: Ordinal10: Segment Accuracy Across Training Steps

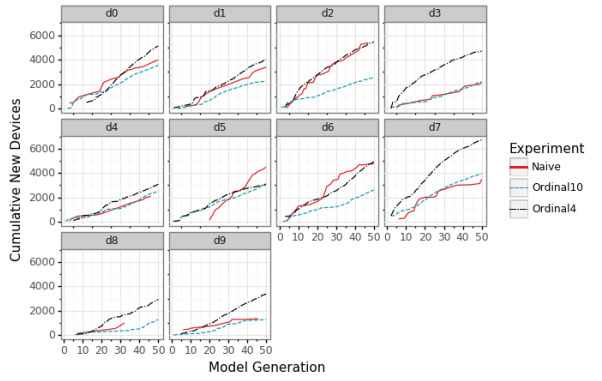

Figure 9: New Devices added in segments with accuracy > 50% vs. Generation

Table 2: Classifier Performance, % Correctly Identified

Experiment	All Observations	High Confidence
Naive	0.3805	0.5754
Ordinal4	0.5378	0.5951
Ordinal10	0.4521	0.6627

split on the data, though as the same data was used with each experiment for both "all observations" and "high confidence", within-experiment differences may best be explained by a combination of class balance changes and randomness within stochastic gradient descent results.

7 CONCLUSIONS AND FUTURE WORK

In this paper we demonstrated that a privacy-sensitive multiclass classifier can be trained with only aggregated ground truth provided

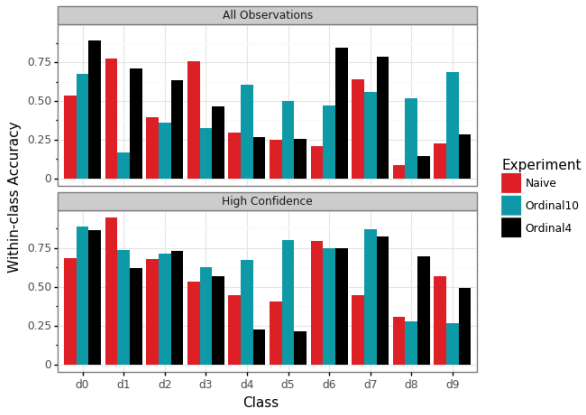

Figure 10: Multiclass Within-Class Performance

by an oracle. In particular, we demonstrated that a quantile classifier acting upon groups of records can learn to create sufficiently homogenous subsets in such a way as to allow the assignment of the crisp, accurate labels required for conventional multiclass learning.

This finding has significant impact in the space of online advertising where concerns over personal privacy, data security and protection of proprietary information may prevent sharing of complete records within or between organizations. Areas of continued study include the application of the various described methods to generate audiences for use in Dstillery's online advertising business as well as to more fully explore the performance of the method under additional constraints including feature sparsity, short-lived observation data and imperfect or noisy oracles. Based upon the promising results shown by the ordinal decile method over the naïve approach we have implemented in production, our engineering group is currently working to upgrade relevant systems to the decile method.

Beyond online advertising, we anticipate our method can provide benefits in other contexts with the ability to repeatedly query for aggregate labels. Cohen *et al.*[**?**] describe particle physics experiments in which only the energy distributions of reactions can be observed, and Rosenman *et al.*[**?**] describe an ecological regression scenario where precinct level voter data is used to predict individual behavior in elections. It is in problems like these where our method may be applied straightforwardly.

ACKNOWLEDGMENTS

We would like to thank the Data Science Research and Machine Learning Engineering teams at Dstillery for supporting us through this process. We would also like to thank Dstillery's technical teams, with particular thanks to Trevor Preston, Matt Sabban, John Scanlan and Ramon Cabral for enabling and supporting our tests as we honed the method. Finally, special thanks to Jay R. Wardle for his artistic contributions to this work.

Algorithm 1: Ordinal Classification

Divide the training data by its labels into q quantiles with $q-1$ cutpoints $\{k_1, k_2, \ldots, k_{q-1}\}$;
for *each cutpoint* k_i **do**
  if *label* $< k_i$ **then**
    relabel as 0;
  else
    relabel as 1;
  end
  Fit binary classifier on new $\{0, 1\}$ labels;
  Compute predicted class membership probabilities $(Pr(target > k_i))$ on the test set;
end
for *each observation in the test set* **do**
  Assign $Pr(V_1) = 1 - Pr(target > k_1)$;
  Assign $Pr(V_i) = Pr(target > k_{i+1}) - Pr(target > k_i), 1 < i < q$;
  Assign $Pr(V_q) = Pr(target > k_{q-1})$;
  Assign class label $q = \text{argmax}_i\, Pr(V_i)$
end

Algorithm 2: Oracle Active Learning

while *Proportion of desired class approaches 1* **do**
  Sort data available for training by label from lowest to highest;
  Select n observations from among the highest labeled observerations;
  Select n observations from among the lowest labeled observations;
  Select n observations at random;
  Fit either **Ordinal Classification** model or **Naïve Regression Model** on observation feature and labels;
  Use trained model to generate predictions on entire dataset universe;
  Sort predictions by predicted label;
  Select top m observations by predicted label;
  for $i \in (1, 2, \ldots, j)$ **do**
    Randomly select a group of observations from top m;
    Submit group to oracle;
    From oracle retrieve an aggregated label describing the group;
    Assign group label value to each observation;
    Re-insert observations into dataset with new label;
  end
end

Algorithm 3: Oracle-Based Multiclass Classification

for $j \in (1, 2, \ldots, M)$ **do**
  for $v \in$ *classes* $\{v_1, v_2, \ldots, v_c\}$ **do**
    Perform **Oracle Active Learning** step;
  end
end
for $v \in$ *classes* $\{v_1, v_2, \ldots, v_c\}$ **do**
  Select observations where class v label close to 1;
end
Combine all selected observations into training set;
Train multiclass classifier;

A ALGORITHMS

REFERENCES

[1] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our Data, Ourselves: Privacy Via Distributed Noise Generation, Vol. 4004. 486–503. https://doi.org/10.1007/11761679_29

[2] Eibe Frank and Mark Hall. 2001. A Simple Approach to Ordinal Classification. In *Machine Learning: ECML 2001 (Lecture Notes in Computer Science)*, Luc De Raedt and Peter Flach (Eds.). Springer, Berlin, Heidelberg, 145–156. https://doi.org/10.1007/3-540-44795-4_13

[3] Wreetabrata Kar, Sarath Swaminathan, and Viswanathan Swaminathan. 2017. Audience Validation in Online Media Using Limited Behavioral Data and Demographic Mix. *International Journal of Semantic Computing* 11, 01 (March 2017), 5–20. https://doi.org/10.1142/S1793351X17400013

[4] Yann LeCun, Corinna Cortes, and CJ Burges. [n.d.]. MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges. http://yann.lecun.com/exdb/mnist/

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[6] Novi Quadrianto, Alex J. Smola, Tiberio S. Caetano, and Quoc V. Le. 2008. Estimating Labels from Label Proportions. In *Proceedings of the 25th International Conference on Machine Learning (ICML '08)*. ACM, New York, NY, USA, 776–783. https://doi.org/10.1145/1390156.1390254

[7] Stewart Renehan. 2019. *Optimizing Demographic Models with External Measurement Source*. Technical Report. PushSpring, Inc. https://docs.google.com/document/d/1XFlGiAuMyHfhfPE9THAljLDoPhs4ghrlyDgWRR65TvU/edit?usp=sharing "Accessed February 2020".

[8] Latanya Sweeney. 2002. k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 05 (Oct. 2002), 557–570. https://doi.org/10.1142/S0218488502001648

[9] Simon Tong. 2001. *Active Learning: Theory and Applications*. Vol. 1. Stanford University.

[10] L.G. Valiant. 1984. A Theory of the Learnable. *Commun. ACM* 27, 11 (1984).

[11] Melinda Williams, Claudia Perlich, Brian Dalessandro, and Foster Provost. 2014. Pleasing the advertising oracle: Probabilistic prediction from sampled, aggregated ground truth. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Aug. 2014). https://doi.org/10.1145/2648584.2648587

8 REPRODUCIBILITY SUPPLEMENT

8.1 Code

All code used in the simulation was posted just as it was run to the author's GitHub at https://github.com/asteriske/privacy_sensitive_active_learning, where instructions are posted to download the code, create the requisite environment and run the simulation.

8.2 Runtime Details

Simulations were run as specified in the published `config_file.yaml`, training for 50 generations with a 60,000 / 10,000 train test split. Ordinal classifier runs of the code ran with `n_ordinal_classes` values of 4 and 10, and `n_worker = 10`. Slower machines may need to increase `data_worker_ttl` and `fitter_worker_ttl` and/or decrease `n_worker` to prevent workers from timing out before all tasks are done.

We ran our simulations on a single server running CentOS Linux 7.4.1708 with 24 Intel Xeon E5-2430L processors at 2.0GHz and 64gb of physical RAM.

Python was run using miniconda 4.8.2, Python version 3.8.1. All packages were as listed in `conda-package-list.txt` in the code repository.